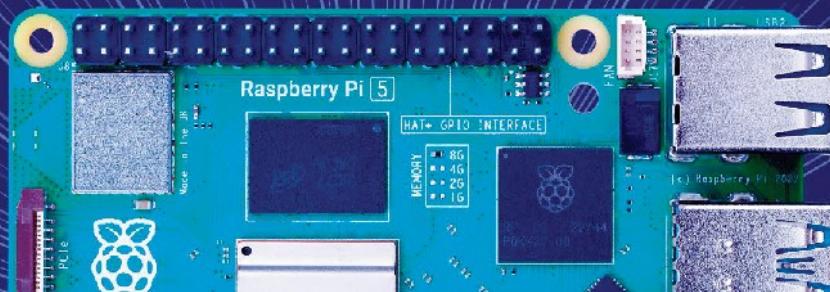
Numero 141 | Maggio 2024 | magpi.cc raspberryitaly.com

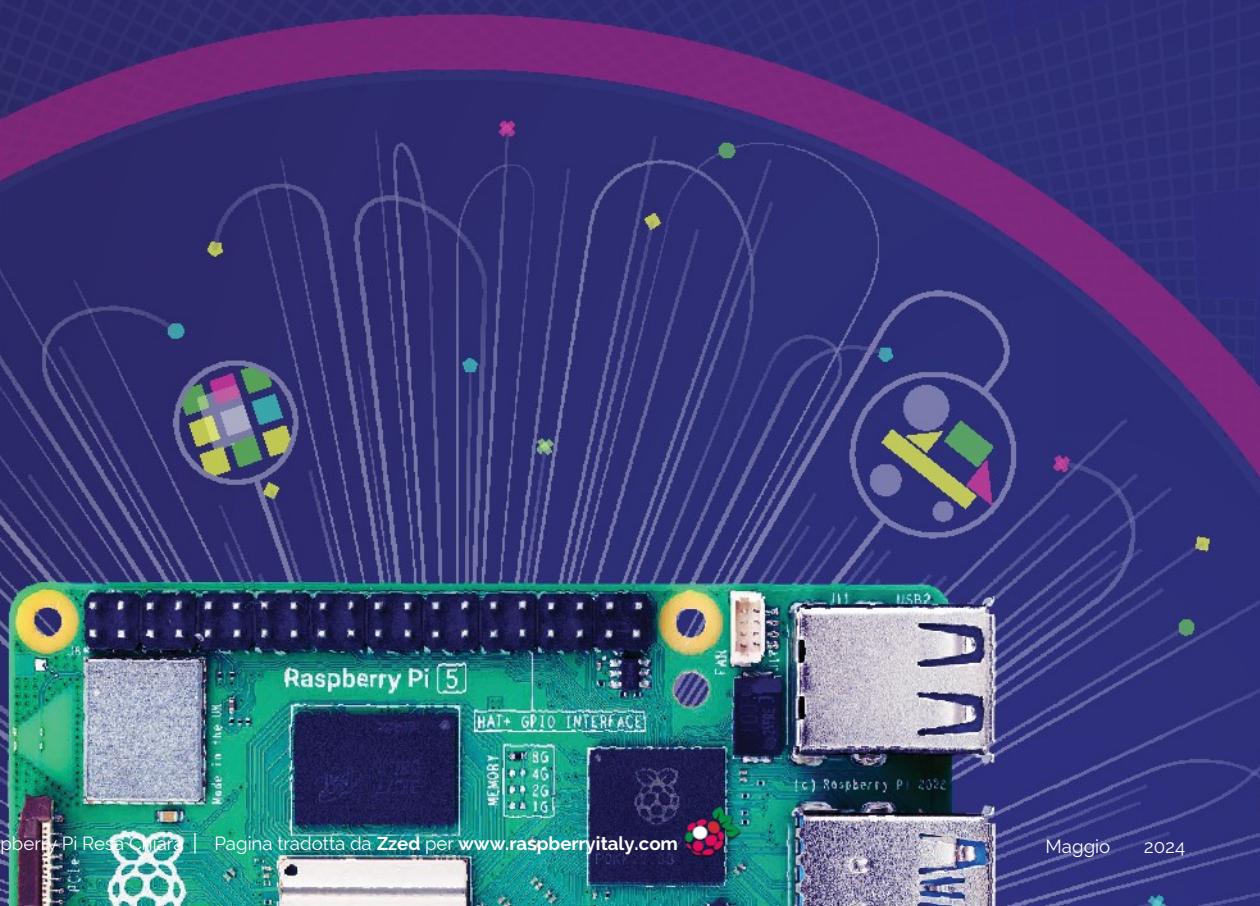
La rivista ufficiale Raspberry Pi tradotta in italiano per RaspberryItaly

IA RASPBERRY PI RESA CHIARA

REALIZZA I TUOI MODELLI

Esplora le funzioni in Python

Test dei case Raspberry Pi 5

Generazione testo e immagini · Sintesi vocale naturale · Telecamere, robot e kit IA

IA RASPBERRY PI RESA CHIARA

GENERA IMMAGINI SENZA VIOLARE IL COPYRIGHT, AGGIUNGI VOCI NATURALI AL TUO LEGGI SCHERMO E INCONTRA UN CANE ROBOT NELLA NOSTRA GUIDA AGLI STRUMENTI DI MACHINE LEARNING PER RASPBERRY PI

MAKER
K.G. Orphanides
K.G. è una scrittrice e sviluppatrice
@owlbear

Il machine learning non è tutto un chiacchierare con i chatbot e generatori di immagini che violano il copyright, ma se avessi pensato così, saresti perdonato, a causa dell'hype attuale sulla intelligenza artificiale. In questo speciale, cercheremo le opzioni migliori per progetti che sfruttano il machine learning, giocattoli divertenti e strumenti veramente utili.

Abbiamo già esaminato la creazione di chatbot (Costruisci il tuo Chatbot GPT, **magpi.cc/129**), utilizzando popolari strumenti online come i modelli GPT non molto open di OpenAI, oppure chatbot che utilizzano modelli completamente locali e incentrati sulla privacy, come quello della Stanford University Alpaca (perfezionato da LLaMA di Meta).

Alcuni LLM (**modelli linguistici di grandi dimensioni**) includono contenuti liberamente accessibili in internet, incluse le pagine web contenenti materiale protetto da copyright, repository GitHub che non hanno rispettato le loro licenze, e sono stati addirittura trovati dati personali – il dataset di allenamento di LLaMA 2 è stato infatti cancellato.

In questo contesto eviteremo modelli con funzionalità problematiche e faremo del nostro meglio per assicurarci di non utilizzare qualsiasi altro modello che contenga spiacevoli sorprese. Abbiamo invece cercato progetti open e di pubblico dominio che possono essere eseguiti localmente sul tuo hardware Raspberry Pi.

Questo approccio ci permette di capirne di più sul machine learning e sugli strumenti di intelligenza artificiale mentre minimizziamo i potenziali danni.

L'etica della IA

Si è parlato molto di un'ipotetica minaccia che una IA generale (il termine popolare per una "vera" intelligenza artificiale - comunque tu scelga di definirla) potrebbe rappresentare per l'esistenza dell'umanità.

Questa è un po' una cortina fumogena per i veri problemi etici con cui gli utenti nel campo del machine learning avranno a che fare: l'elevato consumo energetico necessario per l'allenamento delle IA e nell'utilizzo di modelli basati su cloud; uso insostenibile dell'acqua da parte dei data center dove sono allenate e utilizzate; sfruttamento dei lavoratori sia nel labeling dei dati di allenamento che nella moderazione dei contenuti per sistemi live e per l'abuso di copyright sotto forma di dati, accessibili da internet, usati per addestramento del modello ma senza considerarne i termini della licenza.

Gli utenti sono portati ad aspettarsi risposte autorevoli, e molti sembrano assumere che gli LLM in particolare, conterranno informazioni attuali, anche se non possono riportare nulla di ciò che è accaduto dopo la data dei dati di addestramento del modello.

Ancora più significativo, quel che fanno gli LLM è posizionare le parole in un ordine probabile basato sulla loro formazione: potresti averli sentiti definire "pappagalli stocastici". Agli LLM non vengono forniti dati di formazione etichettati ma piuttosto di una vasta gamma di scritti e di pratiche emulando i modelli in esso contenuti.

Gli LLM creano probabili sequenze di testo, il risultato di un processo formativo in cui il modello si cimenta per prevedere con precisione la parola successiva in un intervallo vasto di testi. Quello che sputano può essere divertente e addirittura impressionante, ma un LLM non può analizzare cosa stai chiedendo, o la sua risposta. Quindi vediamo avvocati citare casi che non esistono, chatbot che sputano fuori storie con falsi ricordi di atti criminali, raccomandano pacchetti software inventati che introducono vulnerabilità nel blockchain o semplicemente che ripetono popolari idee sbagliate e risposte plausibili ma imprecise a sfortunati utenti che li scambiano per un motore di ricerca o per un archivio di informazioni.

> "COME USARE GLI STRUMENTI DI MACHINE LEARNING SENZA ESSERE GOFFAMENTE POCO ETICO"

Riduzione del danno

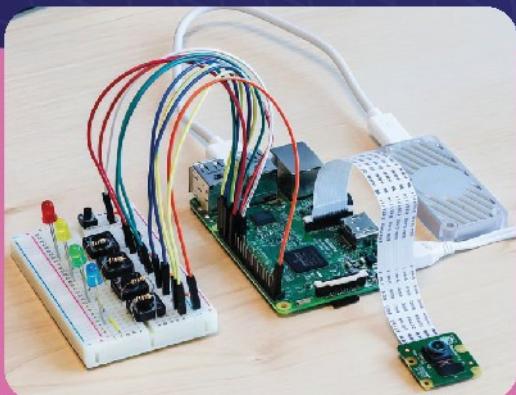
Quindi, come usare gli strumenti di machine learning senza essere goffamente poco etico?

Per cominciare, eseguiremo tutto localmente sul Raspberry Pi. Ciò limita immediatamente il tuo consumo a circa 6W (con Raspberry Pi 5) e ti assicura di poter tenere i tuoi dati per te.

Faremo del nostro meglio per utilizzare modelli e progetti di machine learning che utilizzano solo dati di pubblico dominio o raccolti consensualmente, dati di allenamento, ed evidenziare eventuali problemi con gli altri menzionati. Allo stesso modo daremo priorità ai modelli aperti, dove sono accessibili al pubblico sia i dati che le metodologie di training, piuttosto che scatole nere opache.

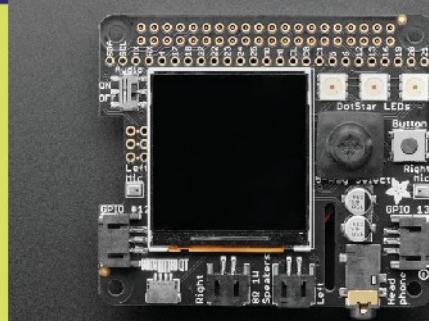
E anche se è difficile evitare di menzionare almeno un paio di colossi tecnologici del machine learning, vogliamo esplorare progetti più importanti, ingegnosità su piccola scala rispetto ai profitti speculativi e con l'influenza delle grandi aziende tecnologiche.

Anche se "intelligenza artificiale" è nella migliore delle ipotesi un termine improprio, le tecnologie che vengono raggruppate sotto questo nome sono spesso affascinanti. Vale anche la pena capire cosa c'è dietro l'ultima mania che sta investendo il settore, e non c'è modo migliore per imparare che fare.

Attenzione!
Copyright

Abbiamo utilizzato modelli liberi da copyright per la generazione delle immagini di questo speciale, ma la maggior parte degli altri modelli diffusi sono stati addestrati con scraping senza permesso e potrebbe compromettere i diritto d'autore di artisti in vita. Leggi la sezione Text e data mining per ricerche non commerciali.

magpi.cc/copyright

PROGETTI IA IN PRIMO PIANO

QUESTE REALIZZAZIONI AIUTANO A PORTARE L'INTELLIGENZA DELLA IA NEL MONDO REALE

Acceleratore Coral USB

magpi.cc/coral
magpi.cc/teachmachine

L'acceleratore USB di Coral ti consente di creare funzionalità di intelligenza artificiale nei progetti Raspberry Pi. L'acceleratore è costruito attorno al chip Edge TPU di Google, un ASIC che accelera notevolmente le prestazioni di rete neurale sul dispositivo.

Abbiamo usato Coral per costruire una Teachable Machine, un dispositivo a cui può essere insegnato a riconoscere gli oggetti, utilizzando Raspberry Pi, Coral USB e una telecamera Raspberry Pi per un sistema completo perfetto per eseguire attività complesse di visione artificiale come il riconoscimento degli oggetti. Poiché l'acceleratore è in funzione localmente, non è necessario connettersi a un cloud o condividere dati protetti su Internet. Funziona anche con una latenza inferiore rispetto a una connessione cloud, eseguendo il rilevamento degli oggetti quasi in tempo reale.

Braincraft HAT di Adafruit

magpi.cc/braincraft

Costruito per Raspberry Pi 4, Braincraft HAT di Adafruit è un kit piuttosto completo per il machine learning. Include uno schermo TFT 240×240 da 1,54 pollici come display, un joystick per navigare tra le opzioni visualizzate su di esso, una ventola di raffreddamento, tre LED controllabili, oltre a porte per altoparlante, cuffie e microfono per aiutarti a creare strumenti di sintesi vocale o progetti di assistenza domestica. Non c'è nessuna fotocamera integrata, ma include uno slot per collegarne una, così come una serie di altri dispositivi IC2 e JST STEMMA.

Adafruit pubblica una serie di progetti mirati a Braincraft e sta attualmente aggiornando le guide e il software per supportare Raspberry PI OS Bookworm. La versione legacy di Raspberry Pi OS Bullseye, che è nota per essere stabile con l'HAT, è disponibile sia tramite Raspberry Pi Imager che tramite download diretto.

> “ I CANI ROBOT XGO MINI 2 E LITE 2 DI LUWU DYNAMICS SEMBRANO LA VERSIONE DA SCRIVANIA DEL ROBOT SPOT DI BOSTON DYNAMICS ”

Vizy Camera

magpi.cc/vizycam

Come XGO, la fotocamera Vizi è gestita da un Raspberry Pi, ma versione standard, non in formato Compute Module. Puoi anche scegliere quanta RAM vuoi perla tua fotocamera.

Una volta configurata sulla tua rete, puoi accedere dal tuo browser alla sua pagina intranet locale. Puoi anche configurarla per essere raggiungibile da internet. Qui hai accesso al feed della videocamera ed eseguire una serie di applicazioni di machine learnig integrate tra cui l'identificazione di oggetti, l'analisi del movimento, un identificatore di specie di uccelli e un rilevatore di animali domestici che può scattare foto o essere utilizzato come trigger per altri eventi. I progetti suggeriti includono un distributore di dolcetti o un lanciatore di palline.

Come ci si aspetterebbe da qualcosa che può essere utilizzato per spiare gli uccelli da giardino, Vizy può diventare impermeabile con una custodia esterna. Altri add-on includono lenti extra, un adattatore 4G per una maggiore portabilità, e un adattatore Power over Ethernet.

Luwu Dynamics XGO 2

magpi.cc/xgo2

I cani robot XGO Mini 2 e Lite 2 di Luwu Dynamics sembrano la versione da scrivania del robot Spot di Boston Dynamics. Il robot può essere indirizzato da una varietà di microcontrollori e computer a scheda singola, ma la configurazione predefinita utilizza Raspberry Pi CM4.

Oltre alle opzioni di controllo manuale, ha un set di routine comportamentali pre-programmate che puoi richiamare, incluso il riconoscimento degli oggetti utilizzando il sistema di rilevamento oggetti in tempo reale Yolo, il riconoscimento dei comandi vocali e si sforza di riconoscere di genere, emozioni e gesti. Tutti questi comportamenti sono gestiti da una serie di script Python e dalle relative librerie, che lo rendono davvero facile da personalizzare.

Se non vuoi tuffarti immediatamente negli script di XGO, sono anche disponibili una serie di interfacce di programmazione e librerie personalizzate per controllare il corpo del cyberdog. Le opzioni spaziano da un'app per smartphone di controllo remoto a un'interfaccia di programmazione a blocchi basata su Python accessibile tramite un browser web.

TRASCRIZIONE E SINTESI VOCALE

Uno degli sviluppi veramente più vantaggiosi derivanti dal machine learning è la capacità di eseguire strumenti offline in grado di fornire trascrizioni testuali sofisticate e accurate di audio parlato e in grado di leggere il testo per te.

Sebbene quest'ultimo abbia purtroppo portato ad un boom di fastidiosi video con voce sintetica, è davvero utile quando è sotto il tuo controllo. Installeremo Speech Note (**magpi.cc/speechnote**), un potente strumento di sintesi vocale neurale text-to-speech (TTS), speech-to-text (STT) e di traduzione automatica che viene eseguito interamente a livello locale per garantire la privacy e configureremo lo screen reader Orca con Piper TTS per dare allo strumento di accessibilità un aspetto più moderno. Per prima cosa configureremo Flatpak e installeremo Speech Note tramite il terminale.

Trascrizione note vocali

```
$ sudo apt install flatpak
$ flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
```

Ora riavvia Raspberry Pi, quindi apri un Terminale e digita:

```
$ flatpak install speechnote
```

Dì "sì" per confermare che desideri utilizzare il riferimento suggerito, che desideri installarlo e desideri procedere con le modifiche suggerite all'installazione del sistema.

Apri SpeechNote dal menu Audio e Video, quindi vai al menu Lingue, cerca italiano (e qualsiasi altra lingua di cui potresti aver bisogno) e seleziona i modelli che desideri utilizzare.

Whisper di OpenAI è open source e gli strumenti di trascrizione non trasportano il fardello della violazione della proprietà intellettuale dei generatori di testo o immagine, quindi English Whisper Small dovrebbe fare la magia per lo Speech To Text (in inglese).

Sono disponibili numerosi modelli di sintesi vocale. Troviamo che Piper Alba Medium sia facile da ascoltare e convenientemente vicino alla cima della lista.

Sono disponibili anche alcune opzioni di traduzione, ma questo elenco è limitato a solo una per coppia linguistica. Puoi provarli in base ai requisiti della tua traduzione: abbiamo installato dall'inglese al francese e dal francese all'inglese: quest'ultimo era disponibile solo quando abbiamo selezionato il francese nel primo passaggio di installazione della lingua. Sono disponibili anche alcuni correttori di punteggiatura, che possono essere utili per gli studenti di lingue.

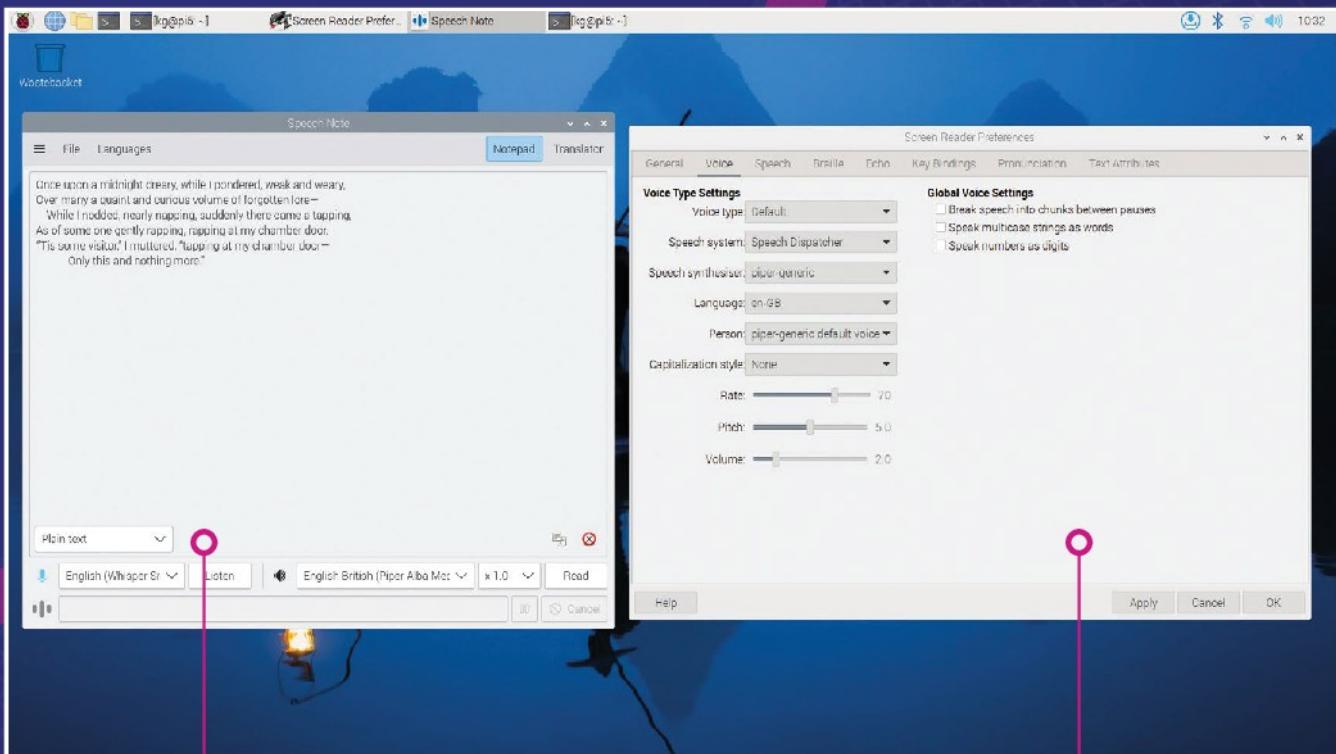
Scarica il seguente file di test, che è un discorso di quattro secondi: **magpi.cc/audiotestwav**.

In Speech Note fai clic su File > Trascrivi un file e apri il file appena scaricato. La trascrizione inizierà automaticamente e il testo verrà visualizzato nel riquadro principale di Speech Note.

Integrare Piper con Orca

La sintesi vocale Piper è un chiaro miglioramento rispetto alle voci sintetizzate eSpeak che tutti conosciamo. Non è ancora disponibile come opzione predefinita per molti lettori di schermo, ma possiamo integrarlo con Orca, lo screen reader Linux più maturo.

Usare Orca con Piper ha i suoi limiti, come le prestazioni di lettura relativamente lente quando si lavora in un terminale e si richiedono più risorse di sistema, ma la voce più naturale può rendere più piacevole lavorare se si interagisce principalmente con una GUI tramite tasti rapidi.

```
$ sudo apt install orca
$ cd Software
$ git clone https://codeberg.org/
MightyOwlbear/rpi-piper-tools.git
$ sh install-spd-piper.sh
```

Inserisci la tua password per installare le dipendenze quando richiesto.

Una volta installato, supponendo che l'audio di prova fosse prodotto, è il momento di configurare il lettore Orca per utilizzare Piper:

```
$ orca -s
```

Nello strumento di configurazione di Orca, seleziona la scheda Voce. In Sintetizzatore vocale, seleziona "voce predefinita generica piper" dal

Speech Note è una applicazione singola, completamente locale per strumenti come trascrizione da Whisper, Piper TTS e traduzioni

Stiamo utilizzando lo screen reader Orca, a cui abbiamo aggiunto Piper neural Text-To-Speech

> "INSTALLEREMO SPEECH NOTE, UNA RETE NEURALE CON SINTESI VOCALE TEXT-TO-SPEECH E TRADUZIONE AUTOMATICA"

menu a discesa.

Deseleziona l'opzione "Dividi il discorso in blocchi tra le pause" nella casella sulla destra, quindi fai clic su Applica. Potresti voler modificare anche la velocità del parlato: ci piace metterla a 70.

Ora sentirai una voce Piper chiamata Alan narrare le tue interazioni. Vale la pena notare che, se sei un dattilografo veloce, Piper non sarà in grado di tenere il passo con il testo inserito lettera per lettera allo stesso modo di eSpeak.

Orca funziona sia con Xorg che con Wayland, ma abbiamo visto prestazioni leggermente migliori in X. Se lo stai avviando dalla riga di comando e avvii la GUI da lì, invoca semplicemente

```
$ startx
```

invece di

```
$ wayfair
```

Se stai avviando la GUI, dovrai cambiare la configurazione per adattarla alle tue preferenze. In un Terminale digita:

```
$ sudo raspi-config
```

Vai a 6 Opzioni avanzate > A6 Wayland > X1 X11 e seleziona OK. Riavvia per rendere effettive le modifiche alle impostazioni.

REALIZZA UN DIFFUSORE DI IMMAGINI CHE RISPETTI L'ARTISTA

Cosa Serve

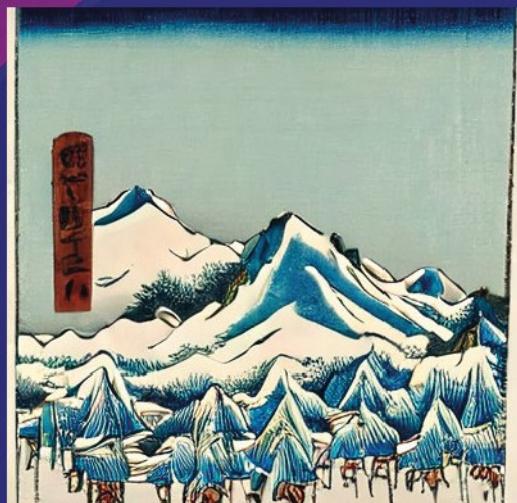
- 8GB RAM (Nota che potrebbe non funzionare sui modelli da 4GB)
- Scheda microSD capiente (minimo 64GB)
- Mitsua Diffusion One **magpi.cc/mitsua**
- Python 3.10.6 (Nota: Le versioni più recenti non supportano PyTorch)

La generazione di immagini è un campo minato se speri di mantenere l'uso del machine learning etico ed economico, oltre che divertente. Servizi come Midjourney e Dall-e sono scatole nere per gli utenti (non puoi vedere cosa succede al loro interno). Acquistare tempo su questi servizi è costoso sia per te, che per l'ambiente in termini non solo, in primo luogo, di addestramento dei modelli, ma anche di alimentazione e raffreddamento dei data center dove vengono eseguiti.

Molti generatori di immagini si basano anche sul lavoro di artisti viventi, senza il loro consenso, rendendoli un grosso problema. Fortunatamente, stiamo iniziando a veder emergere modelli di generazione di immagini basati su materiale di pubblico dominio come gallerie di immagini condivise da musei e opere d'arte offerte dai loro autori volontariamente e preventivamente.

▲ **Prompt**: "un paesaggio impressionista nello stile di Turner, i raggi di sole sono visibili attraverso le nuvole, illuminando la campagna, alberi, fiume". **Prompt negativo**: "persone, animali"

> "LA GENERAZIONE DI IMMAGINI È UN CAMPO MINATO SE SPERI DI MANTENERE L'UTILIZZO DEL ML ETICO ED ECONOMICO"

Per questo tutorial, abbiamo scelto Mitsua Diffusion One (**magpi.cc/mitsua**), un modello text-to-image compatibile con Stable Diffusion addestrato esclusivamente con contenuti di pubblico dominio. La sua licenza richiede di non violare il diritto d'autore nella messa a punto del modello e di non spacciare le immagini generate come creazioni non basate sull'AI.

La libreria Diffusers di Hugging Face ci consente di eseguire modelli di diffusione – che funzionano riducendo il rumore finché non appare un' immagine che soddisfa il prompt – su quasi tutte le CPU. Noi stiamo utilizzando Pi 5.

▶ **Prompt**: "stampa xilografica ukiyo e del Monte Fuji, ricoperto con la neve, nello stile di Hokusai". **Prompt negativo:** "persone, animali"

▼ **Prompt**: "un dipinto dell'antico enorme tempio giapponese di Kyoto, tempio di Kiyomizu, un dipinto impressionista, belle arti". **Prompt negativo**: "foto"

▼ **Prompt**: "una piovosa strada britannica, belle arti, John Atkinson Grimshaw, autunno, alberi, sera". **Prompt negativo**: persone, animali

▲ **Prompt**: "cielo notturno nello stile di Van Gogh. Le case della città sono visibili sotto

▲ **Prompt**:"un dipinto astratto nello stile di Mondrian, con linee nere che segnano forme quadrate e rettangolari su uno sfondo bianco, alcuni riempiti di colore rosso, blu e nero. pittura ad olio". **Prompt negativo**: "umani, animali, curve"

01 Imposta il tuo disco

Avrai bisogno di un Raspberry Pi 5 da 8 GB (o di Raspberry Pi 4) con una scheda microSD da 64 GB. Su una nuova installazione di Bookworm: lo useremo headless, quindi puoi disabilitare l'avvio della GUI o installare la versione Lite di Raspberry Pi OS. Prima di masterizzare l'immagine del OS su una scheda microSD con Raspberry Pi Imager, ricorda di abilitare SSH con l'autenticazione tramite password, imposta un nome host, un nome utente e la password e configura la LAN wireless se ne hai bisogno.

Se hai installato la versione completa di Raspberry Pi OS con GUI, ti consigliamo disabilitarla. Accedi tramite SSH o apri un Terminale e digita:

```
$ sudo raspi-config
1 System Options > S5 Boot / Auto Login > B2
Console Autologin
```

Selezione Fine e premi Yes per riavviare

02 Configura lo swap

Se non hai almeno 8 GB di spazio swap – spazio di lavoro aggiuntivo sulla scheda microSD per quando si esaurisce la RAM – Diffusers fallirà nel riprodurre l'immagine dopo l'elaborazione. Tieni presente che questo può consumare la scheda microSD più velocemente di quanto ci si possa aspettare.

```
$ sudo dphys-swapfile swapoff
$ sudo nano /etc/dphys-swapfile
```

In nano, modificheremo le voci swapsize e maxswap per darci 8 GB di swap e nessun limite sulla dimensione dello swap:

```
CONF_SWAPSIZE=8192
CONF_MAXSWAP=
```

Premi **CTRL** + **X** e **Y**. Quindi, torna al prompt dei comandi:

```
$ sudo dphys-swapfile setup
$ sudo dphys-swapfile swapon
$ reboot
```

03 Clonazione su larga scala

Ora, alcune dipendenze prima di iniziare. Innanzitutto, dovremo aggiungere a Git il supporto per file di grandi dimensioni:

```
$ curl -s https://packagecloud.io/install/
repositories/github/git-lfs/script.deb.sh |
sudo bash
$ sudo apt install git-lfs
```

Inizializzalo:

```
$ git-lfs install
```

Crea una directory per il nostro modello e scaricalo:

```
$ mkdir -p Software/models && cd Software/
models
$ git clone https://huggingface.co/Mitsua/
mitsua-diffusion-one
```

Sebbene non particolarmente grande per gli standard dei modelli di diffusione, Mitsua Diffusion One ha comunque una stazza di circa 6 GB, quindi questa operazione potrebbe richiedere del tempo.

04 Devi tenerli separati

Per utilizzare Hugging Face Diffusers, avrai bisogno di PyTorch, che richiede Python nella versione massima di 3.10.6, piuttosto vecchia rispetto all'attuale versione. Utilizzeremo pyenv per installarlo e il comando virtualenv per assicurarci di bloccare il nostro ambiente di lavoro sulla giusta versione di Python e mantenere tutte le dipendenze rilevanti isolate. Ciò garantisce che nulla di ciò che facciamo qui avrà un impatto su qualsiasi altra cosa abbiamo installato o sulle impostazioni predefinite del nostro software a livello di sistema.

```
$ cd
```

```
$ curl https://pyenv.run | bash
```

Ti verrà richiesto di aggiungere scorciatoie al tuo profilo bash per richiamare pyenv dalla riga di comando.

05 Aggiorna il tuo profilo

Nel Terminale, digita:

```
$ nano ~/.bashrc
```

Aggiungi il seguente blocco alla fine del file e salva con CTRL+X.

```
echo ‘export PYENV_ROOT=”$HOME/.pyenv”’ >>
~/.bashrc
echo ‘command -v pyenv >/dev/null || export
PATH=”$PYENV_ROOT/bin:$PATH”’ >> ~/.bashrc
echo ‘eval “$(pyenv init -)”’ >> ~/.bashrc
```

Ora modifica il file .profile:

```
$ nano ~/.profile
```

Come sopra, aggiungi questo blocco alla fine del file e salva:

```
echo ‘export PYENV_ROOT=”$HOME/.pyenv”’ >>
~/.profile
echo ‘command -v pyenv >/dev/null || export
PATH=”$PYENV_ROOT/bin:$PATH”’ >> ~/.profile
echo ‘eval “$(pyenv init -)”’ >> ~/.profile
```

Ora ricarica il tuo profilo bash, così:

```
$source ~/.bashrc
```

06 Python dipende da te

Utilizzeremo pyvenv per installare la versione giusta di Python, ma abbiamo alcune dipendenze da installare per prime. Le abbiamo divise in tre commenti per facilitare la copia, ma puoi unirli in un unico comando apt install se preferisci.

```
$ sudo apt install zlib1g zlib1g-dev libssl-dev
libbz2-dev libsqlite3-dev
$ sudo apt install libedit-dev libncurses5-dev
liblzma-dev
$ sudo apt-get install libreadline-dev libffi-dev
```

E adesso possiamo installare la versione corretta di Python, gestito da pyenv.

```
$ pyenv install 3.10.6
```

07 Ambiente virtuale

Un altro passo importante per mantenere separate più configurazioni Python: creeremo un ambiente virtuale, in cui installeremo

```
$ pyenv virtualenv 3.10.6 venv_diffusion
```

Ora lo attiveremo. Dovrai ripetere questo passaggio ogni volta che desideri utilizzare il nostro script per diffusion.

```
$ pyenv activate venv_diffusion
```

Il prompt dei comandi verrà ora contrassegnato (**venv_diffusion**). Installeremo il software di cui abbiamo bisogno per invocare il nostro modello.

```
$ cd Software
$ git clone https://codeberg.org/MightyOwlbear/
Raspberry-Pi-5-Diffusion.git
$ cd Raspberry-Pi-5-Diffusion
$ nano python mitsua_app.py
```

08 Quasi arrivati

Immetti:

```
$ cd Software
$ git clone https://codeberg.org/MightyOwlbear/
Raspberry-Pi-5-Diffusion.git
$ cd Raspberry-Pi-5-Diffusion
$ nano python mitsua_app.py
```

Top Tip

Non scordare di attivare

Digita **pyenv activate venv_diffusion** per attivare il tuo ambiente virtuale.

Alla riga 9, controlla che il percorso corrisponda a quello dove hai scaricato il modello Mitsua Diffusion One. Nel nostro script di esempio, abbiamo lavorato con il nome utente "pi", altrimenti adatta la directory riportata nel tutorial. Premi **CTRL+X** per uscire e salvare quando richiesto.

```
$ python mitsua_app.py
```

Ti verrà chiesto di fornire un prompt per il modello: una descrizione dell'immagine che vorresti che producesse. Ti verranno quindi richiesti suggerimenti negativi: quello che non vuoi che l'immagine includa: un oggetto, un colore, una caratteristica come la sfocatura o dita fuse o uno stile, ad esempio fotografia.

La prima volta che esegui questo script, impiegherà più tempo. Se tutto è configurato correttamente, tuttavia, dovrebbe durare meno di 15 minuti e produrrà un file di output con timestamp.

09 Linee, please

Poiché è istruito con materiale di pubblico dominio, principalmente da musei e gallerie d'arte, Wikimedia Commons e sottoinsiemi di pubblico dominio di vari siti di condivisione di immagini, Mitusa Diffusion One ha punti di forza e di debolezza specifici.

Avrai più fortuna se gli fai produrre immagini alquanto convincenti di xilografie, incisioni, impressionismo e dipinti d'arte piuttosto che fotografie. Non siamo mai riusciti a avere una immagine davvero convincente di un gatto.

Per ottenere i migliori risultati, abbiamo trovato utile nominare artisti e stili specifici. Visto che gran parte dei i suoi dati di addestramento provengono da collezioni di gallerie d'arte, vale la pena dare un'occhiata, ad esempio, all'Art Institute of Chicago's collection (**magpi.cc/aicpub**) per trovare qualche ispirazione.

In questo articolo, troverai esempi di stimoli e le immagini da essi prodotte.

Genera descrizioni di immagini

twinery.org

La descrizione dell'immagine è incredibilmente importante nell'internet moderna. Il testo alternativo dell'immagine è indispensabile innanzitutto per gli utenti non vedenti e ipovedenti, ma anche per chiunque interagisca con il Web utilizzando testo, inclusi feed RSS e browser testuali come Lynx.

Sebbene il testo alternativo realizzato a mano è meglio di qualcosa di automatizzato, non tutti trovano facile scrivere. E a volte potresti aver bisogno di uno strumento per descrivere rapidamente un'immagine per te per le tue ragioni. Questo progetto utilizza la stessa venv Python e la stessa struttura di directory che abbiamo impostato per il nostro progetto di generazione di immagini sicure da copyright e utilizza il modello di descrizione dell'immagine BLIP di Salesforce (Bootstrapping Language-Image Pre-training) (**magpi.cc/blip**).

Se preferisci non utilizzare gli strumenti di machine learning dei giganti della tecnologia, preferirai evitare Salesforce. Un documento che descrive lo sviluppo di BLIP (**magpi.cc/blippdf**) indica che il suo allenamento è basato su coppie di immagini e testo alternativo recuperate dal web, poi filtrate per evitare didascalie "rumorose". Il processo di filtraggio e la limitazione della lunghezza delle didascalie, minimizza la probabilità di generare violazioni del copyright, ma la metodologia di allenamento dovrebbe essere presa in considerazione se intendi utilizzare le didascalie generate da BLIP.

Inizieremo scaricando il modello e lo script image description.

```
$ cd ~/Software/models
$ git-lfs install
$ git clone https://huggingface.co/
Salesforce/blip-image-captioning-base
$ cd ..
$ git clone https://codeberg.org/
MightyOwlbear/blip-image-description-tool.git
$ cd blip-image-description-tool
$ nano captioner.py
```

Controlla i percorsi per assicurarti che corrispondano alla posizione del tuo modello: se il tuo utente non si chiama "pi", dovrai cambiarli. Premi **CTRL + X** per salvare e uscire da nano.

```
$ pyenv activate venv_diffusion
$ python captioner.py
```

Questa versione accetta solo indirizzi web per i suoi target immagine. Per testarlo, puoi usare il seguente URL di immagine: **magpi.cc/gullsjpg**. Ddovrebbe produrre una descrizione che reciti: "un gruppo di uccelli che volano nel cielo".

mitsua-app.py

SCARICA IL CODICE COMPLETO

> Linguaggio: **Python**

```
001.  from diffusers import StableDiffusionPipeline
002.  from PIL import Image
003.  import datetime
004.
005.  prompt = input("Descrivi l'immagine che vuoi: ")
006.  negative_prompt = input("Descrivi i tratti da evitare - ad
      es. mani sfocate e fuse: ")
007.
008.  # inserisci il percorso per Mitsua Diffusion One di seguito
009.  pipe = StableDiffusionPipeline.from_pretrained("/home/pi/
      Software/models/mitsua-diffusion-one/",
      low_cpu_mem_usage=True)
010.  pipe = pipe.to("cpu")
011.
012.  image = pipe(prompt, negative_prompt=negative_prompt,
      num_inference_steps=31, width=400, height=400).images[0]
013.
014.  # codificare il timecode di ogni immagine di output in
      mododo che non possano sovrascriversi
015.  created = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
016.  image.save("output-" + created + ".png")
```

Top Tip

htop

Se hai uno schermo collegato al Raspberry Pi, puoi lanciare htop per monitorare i consumi di memoria e swap

captioner.py

SCARICA IL CODICE COMPLETO

magpi.cc/bliptool

> Linguaggio: **Python**

```
001.  import requests
002.  from PIL import Image
003.  from transformers import BlipProcessor,
      BlipForConditionalGeneration
004.
005.  nserisci il percorso della directory del tuo modello in
      entrambe le righe seguenti
006.  processor = BlipProcessor.from_pretrained("/home/pi/Software/
      models/blip-image-captioning-base/")
007.  model = BlipForConditionalGeneration.from_pretrained("/home/
      pi/Software/models/blip-image-captioning-base/")
008.
009.  img_url = input("Digita URL immagine JPG: ")
010.  raw_image = Image.open(
      requests.get(img_url, stream=True).raw).convert('RGB')
011.
012.  inputs = processor(raw_image, return_tensors="pt")
013.
014.  out = model.generate(**inputs, max_new_tokens=1000)
015.  print(processor.decode(out[0], skip_special_tokens=True))
```

Parte 02

Imparare Python: esplora le funzioni e crea una app CLI

Diventiamo funzionali con Python. In questo tutorial imparerai come definire e chiamare le funzioni per evitare di scrivere lo stesso codice più e più volte

MAKER

Lucy Hattersley

Lucy è capo redattrice di *The MagPi* ed è per lo più funzionale.

magpi.cc

Le funzioni sono il cuore pulsante di Python. Ti aiutano a scomporre i programmi Python in piccole parti gestibili che possono essere facilmente ripetibili. Rendono il tuo programma più modulare e più facile da capire.

Ad alcuni piace pensare alle funzioni come taglia e incolla. Scrivi il codice una volta ("definire" la funzione) e quindi incolli il codice ("chiamare" la funzione). Anche se questo è utile per cominciare, le funzioni sono più potenti del taglia e incolla perché accettano input e forniscono output.

In questo tutorial, esamineremo la definizione e la chiamata di funzioni con argomenti diversi. Quindi le utilizzeremo per creare una versione avanzata del programma ToDo dell'ultimo numero che salva una lista in un file e può essere eseguito da qualsiasi punto della CLI (Interfaccia della riga di comando).

> Le funzioni sono il cuore pulsante di Python

Le funzioni vengono definite utilizzando il comando **`def`**, seguito dal nome della funzione e dalle parentesi (che contengono parametri opzionali da passare alla funzione). Un comando **`return`** opzionale rimanda tutti i risultati della funzione al programma principale. Qualcosa tipo:

```
def funzione(parametri_opzionali):
        # codice che fa cose
        return optional_return_value
```

Il codice sotto una funzione è rientrato di quattro spazi. Questo fa sapere a Python che il codice rientrato viene eseguito quando viene utilizzata la funzione (nota come "chiamata di funzione").

Creiamo una funzione reale. Apri Thonny IDE e crea un nuovo file (salvalo come **add_numbers.py**. Ora inserisci queste righe:

```
# Definisci una funzione che prende due numeri
come argomenti e ritorna la loro somma
def somma_numeri(num1, num2):
    totale = num1 + num2
    return totale

# Richiama la funzione con 2 e 3 come argomenti
result = somma_numeri(2, 3)
print(f"LA somma è: {result}")
```

Cosa Serve

- Raspberry Pi
- Raspberry Pi OS
- Thonny IDE

> Il codice compreso in una funzione è rientrato di quattro spazi

La funzione di questo programma è di prendere due numeri e sommarli tra loro. Abbastanza semplice. Il programma è diviso in due parti. La prima parte è il **`def`** che definisce la funzione:

```
def add_numbers(num1, num2)
```

La seconda parte è la chiamata alla funzione, che chiede alla funzione **`add_numbers`** di sommare i numeri 2 e 3 e memorizzare la variabile **`total`** restituita in un'altra variabile chiamata **`result`**.

```
result = add_numbers(2, 3)
```

Infine, stampiamo la risposta utilizzando **`print()`**. Fai clic su Esegui in Thonny e dovresti vedere: "La somma è: 5" nella finestra della Shell sottostante.

REPL Rispondi al mittente

Una delle gioie di Python è che puoi interagire con i programmi nella Shell durante l'esecuzione grazie a REPL (Read-Eval-Print Loop). Fai clic sulla Shell e chiama direttamente la funzione con tutti i valori che desideri.

```
add_numbers(6, 2)
```

Ciò restituisce 8 nella Shell. Mentre siamo in REPL dovremmo parlare di "ambito" o "ambito della funzione" in questo particolare concetto. In REPL puoi accedere e guardare direttamente le variabili del programma. Immettere **`result`** e la Shell visualizzerà "5" (il valore memorizzato dal programma nella riga 7 durante la chiamata della funzione, quando restituisce il valore totale).

Tuttavia, inserisci **`total`**, e la Shell restituisce:

```
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
```

La nostra introduzione alle funzioni prende due numeri e li somma tra loro

```
NameError: name 'total' is not defined
```

Perché possiamo accedere alla variabile **`result`**, ma non alla variabile **`total`**? La risposta è che le variabili all'interno della definizione (la parte rientrata sotto **`def`**) sono "locali". Ciò significa che sono accessibili solo localmente all'interno di quella funzione.

Una volta fuori dalla funzione e nel tuo programma le variabili locali (come la nostra **`total`**) non sono più necessarie, e in un ambiente di programmazione moderno come Python lo spazio di memoria viene pulito e riutilizzato.

Questo è il motivo per cui è importante utilizzare **`return`** e memorizzare eventuali variabili o output della funzione che desideri utilizzare nel resto del codice.

Top Tip

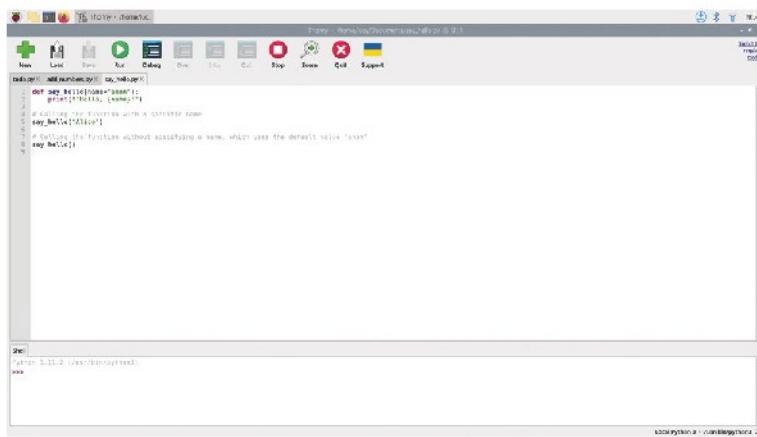
Parametri vs Argomenti

I parametri sono le variabili che sono elencate all'interno delle parentesi nella definizione della funzione. Gli argomenti sono i valori passati alla funzione quando viene chiamata.

Variabili di default

Solleviamo un altro problema nella Shell: la chiamata alla funzione **`add_numbers()`** senza argomenti:

```
add_numbers()
```

La Shell restituisce un altro messaggio di errore:

```
TypeError: add_numbers() missing 2 required
positional arguments: 'num1' and 'num2'
```

La lettura del messaggio di errore ci dice cosa non funziona. "2 argomenti posizionali richiesti mancanti" (i nostri due numeri).

Possiamo risolvere questo problema aggiungendo due numeri alla chiamata di funzione. Ma è anche possibile aggiungere parametri predefiniti nella definizione di funzione.

Supponiamo che stiamo creando un programma che si rivolge alle persone per nome, ma permette loro di restare anonime. Potrebbe dire "ciao" quando inseriscono il loro nome

```
def say_hello(name):
    print(f"Ciao, {name}")
```

Potremmo chiamare questa funzione usando:

```
say_hello("Alice")
```

Ma se vogliamo includere il predefinito anonimo lo aggiungiamo ai parametri usando **`name="anon"`**. Inserisci il codice da **say_hello.py**.

```
def say_hello(name="anon"):
    print(f"Ciao, {name}!")

# Chiana la funzione con un nome specifico
say_hello("Alice")

# Chiamata di funzione senza specificare un
nome, utilizzando il valore predefinito "anon"
say_hello()
```

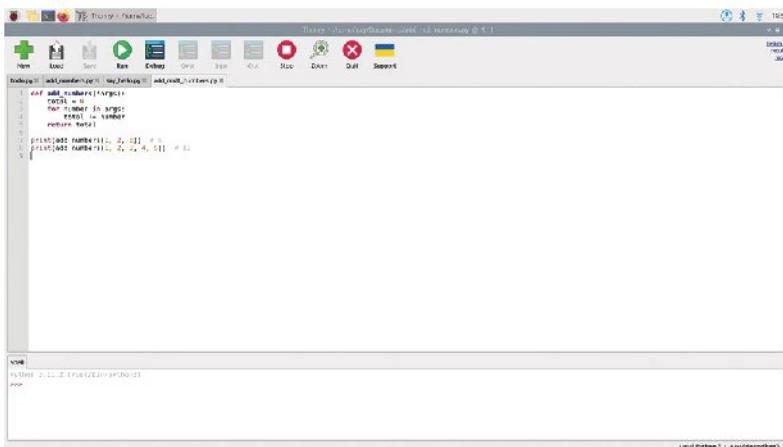
Eseguilo e vedrai:

```
Ciao, Alice!
Ciao, anon!
```

Questa funzione ha un valore predefinito in caso manchi un argomento. In questo esempio, usa 'anon' invece di un nome fornito

Approfondire gli argomenti

Uno dei limiti delle funzioni che abbiamo creato finora è che funzionano solo con un numero specifico di argomenti. Il nostro programma **add_numbers.py** somma solo due numeri tra loro,per esempio.

E se volessi gestire più argomenti di importi diversi? In questo caso è necessario conoscere il parametro **`*args`**.

Questo è un tipo speciale di parametro che gestisce più argomenti. Il "*" dice a Python di metterli in un tipo speciale di elenco chiamato "tupla". Crea ed esegui il seguente programma, chiamandolo **add_mult_numbers.py**.

```
def add_numbers(*args):
    total = 0
    for number in args:
        total += number
    return total

print(add_numbers(1, 2, 3))  # 6
print(add_numbers(1, 2, 3, 4, 5))  # 15
```

Eseguilo e la Shell restituisce 6 e 15. C'è anche un secondo parametro speciale chiamato **`**kwargs`** che

Questa funzione richiede molteplici argomenti utilizzando il parametro *args invece di specificare i parametri individualmente

Top Tip

Cos'è una tupla

Una tupla è un tipo speciale di dati strutturati simile a un elenco, ma immutabile. Una volta creato, non può essere cambiato, solo utilizzato o cancellato.

gestisce più coppie chiave/valore (di cui non parleremo qui poiché le coppie chiave/valore sarebbero un argomento che distrae). Ci sono due valori da decomprimere, da qui i due asterischi.

Ricorsione

Fai un respiro profondo, perché è probabile che arrivi il mal di testa. La ricorsione è un concetto di programmazione relativamente complicato che, se ci pensi troppo, provoca rapidamente mal di testa.

Il concetto generale è quello di una definizione di funzione contenente una chiamata a se stessa. Quando la funzione è chiamata, a sua volta, richiama un'altra versione di se stessa con un argomento modificato.

Una semplice funzione ricorsiva è questa, quella che calcola la somma di tutti i numeri fino a un dato numero. Ad esempio, per trovare la somma di tutti numeri fino a 4, calcolerai 4 + 3 + 2 + 1 = 10.

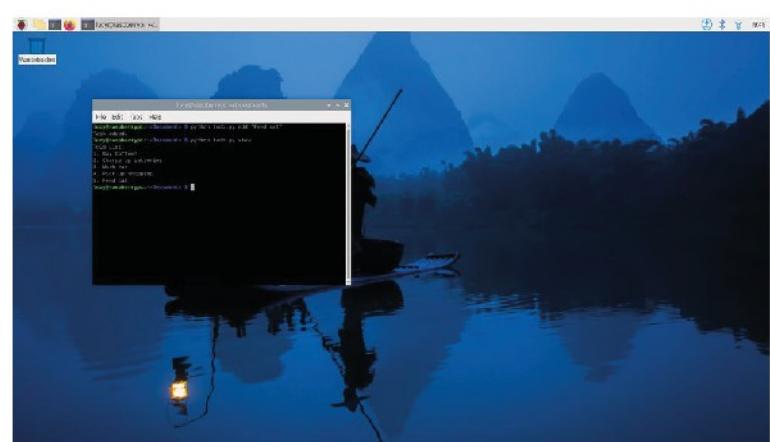
```
def sum_up_to(n):
    # Caso base: se n è 0, la somma fino a 0
is just 0
    if n == 0:
        return 0
    # Rcaso ricorsivo: n più la somma dei
numeri fino a n-1
    else:
        return n + sum_up_to(n-1)

# Esempio d'utilizzo
print(sum_up_to(4))  # Output: 10
```

Questa funzione lavora prendendo il numero intero fornito (come 'n') e vi somma tutti i numeri inferiori ad esso. Lo fa aggiungendo n alla somma di n-1 (chiamando sum_up_to(n-1) per tutti i numeri finché non arriva a 0.

Quindi risale la catena per produrre la somma finale. La ricorsione è una di quelle parti magiche di Python che incontrerai abbastanza frequentemente, e funziona, ma può essere piuttosto impegnativa da visualizzare in mente. Non preoccuparti se non capisci tutto da questa semplice introduzione. Ci sarà tempo, in seguito, per comprenderla.

Riga di comando

Trasformiamo ciò che abbiamo imparato sulle funzioni in un programma utile. Espanderemo la ToDo list dell'esercizio precedente. Questa volta lo trasformeremo in un programma più dettagliato che viene eseguito dalla riga di comando.

Inserisci il codice da **todo.py** in Thonny e salvalo sul disco. Apri una finestra del Terminale, vai alla stessa directory del programma **todo.py** ed esegui il programma per aggiungere, visualizzare ed eliminare le attività.

- Per aggiungere un task, usa: **`python todo.py add "Qui i dettagli del Task"`**
- Per vedere tutti i tasks, usa: **`python todo.py view`**
- Per eliminare un'attività tramite il suo numero (come mostrato in view), usa: **`python todo.py delete 1`** (dove 1 è il numero dell'attività da eliminare)

Questa applicazione dimostra la gestione di base, l'analisi degli argomenti e le semplici operazioni CRUD (Create, Read, Update, Delete) in una

Il programma Python todo.py eseguito dal terminale

> Abbiamo rimosso l'estensione .py spostando il file. Così possiamo eseguire il comando da qualsiasi punto del Terminale

Top Tip

Esegui da Thonny

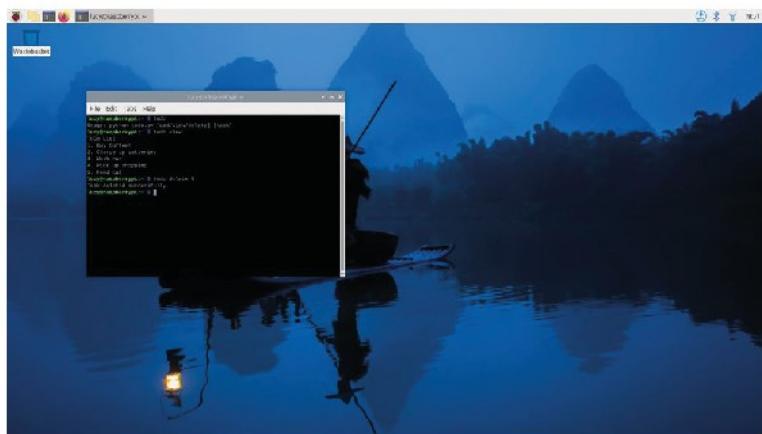
Puoi eseguire **todo.py** da Thonny con gli argomenti utilizzando il comando **`%Run`** nella Shell, ad esempio **`%Run todo.py view`**.

applicazione a riga di comando utilizzando Python. Sentiti libero di ampliarlo con più funzionalità, come le attività di modifica o categorizzazione.

Scriviamolo

Per trasformare **`todo.py`** in un'applicazione (o script) a tutti gli effetti, che puoi eseguire dalla riga di comando, segui questi passaggi.

Aggiungi una riga "shebang" (**`#!`**) all'inizio del programma. Questo dice alla riga di comando quale versione di Python utilizzare. Assicurati che questa sia la primissima riga.

```
#!/usr/bin/env python3
```

Ora rendi eseguibile lo script. Apri una finestra del Terminale nella stessa directory di **todo_list.py** e inserisci quanto segue:

```
cd /path/to/todo.py
chmod +x ./todo.py
```

Sostituisci **/path/to/todo.py** con la tua directory. Nel nostro caso: **/home/lucy/Documents/todo.py**.

Il comando **`chmod +x`** aggiunge diritti di eseguibilità al file, consentendo l'esecuzione del codice dalla riga di comando.

Il programma ToDo finale in esecuzione direttamente come comando da qualsiasi Terminale

Ora copia il file dalla tua directory di lavoro in una directory nel tuo PATH. Il path è un elenco delle posizioni in cui il terminale controlla i file quando esegui i comandi nella shell. Puoi vedere le posizioni attuali del PATH con:

```
echo $PATH
```

Le directory nel PATH sono elencate con i due punti ":" che le separano. Un luogo comune in cui spostare i file è **/usr/local/bin** (bin sta per "binario"). Copieremo qui il file **todo.py** (ti servirà usare **`sudo`** per ottenere l'autorizzazione):

```
sudo cp ./todo.py /usr/local/bin/todo
```

Nota che abbiamo rimosso l'estensione .py quando si sposta il file. È così che possiamo semplicemente eseguire il comando da qualsiasi punto del Terminale utilizzando "todo" come faremmo con qualsiasi altro comando come ls, cp, o cat. Ora chiudi il Terminale e apri un'altra finestra o immetti il seguente comando per rinfrescare la shell.

```
source ~/.bashrc
```

Ora puoi eseguire il programma da qualsiasi luogo nel Terminale utilizzando solo il comando todo seguito da un'opzione:

- todo add
- todo view
- todo delete

Congratulazioni, ora hai un programma ToDo funzionante che salva l'elenco delle cose da fare in un file chiamato todo_list.txt nella tua directory home. Puoi visualizzare ToDo utilizzando l'opzione di visualizzazione delle cose da fare o utilizza **`cat ./todo_list.txt`**.

C'è molto altro che puoi fare con il tuo programma ToDo. Soprattutto speriamo che ti sia piaciuto conoscere le funzioni e costruire un programma completo funzionale che si aggiunge al tuo Terminale di Raspberry Pi.

todo.py

SCARICA IL CODICE COMPLETO: magpi.cc/github

Linguaggio: **Python**

```
001. #!/usr/bin/env python3
002.
003. import sys # Importa il modulo sys per accedere
     agli argomenti da riga di comando
004. import os # Importa il modulo os per verificare se
     un file esiste
005.
006. # Definisce il percorso del file
007. home_dir = os.path.expanduser('~')  # Prende la home
     directory dell’utente
008. todo_file = os.path.join(home_dir,
     'todo_list.txt')  # Costruisce il percorso completo
009.
010. def load_tasks():
011.     # Carica i compiti dal file
012.     if os.path.exists(todo_file):
013.         with open(todo_file, 'r') as file:
014.             tasks = file.readlines()
015.             tasks = [task.strip() for task in
     tasks]  # Rimuove il carattere di nuova riga
016.     else:
017.         tasks = [] # Ritorna una lista vuota se il
     file non esiste
018.     return tasks
019.
020. def save_tasks(tasks):
021.     #Salva i compiti nel file
022.     with open(todo_file, 'w') as file:
023.         for task in tasks:
024.             file.write(f"{task}\n")
025.
026. def add_task(task):
027.     # Aggiungere un nuovo compito
028.     tasks = load_tasks()
029.     tasks.append(task)
030.     save_tasks(tasks)
031.
032. def view_tasks():
033.     # Mostra tutti i compiti
034.     tasks = load_tasks()
035.     if tasks:
036.         print("Lista delle cose da fare:")
037.         for idx, task in enumerate(tasks,
     start=1):
038.             print(f"{idx}. {task}")
039.     else:
040.         print("La tua lista è vuota!")
041.
042. def delete_task(task_number):
043.     # Cancella un compito con il suo numero
044.     tasks = load_tasks()
045.     if 0 < task_number <= len(tasks):
046.         del tasks[task_number - 1]
047.         save_tasks(tasks)
048.         print("Voce cancellata con successo.")
049.     else:
050.         print("Numero compito non valido.")
051.
052. def main():
053.     # Check if the user has provided a command
054.     if len(sys.argv) < 2: # The first argument
     is the script name
055.         print("Sintassi: python todo.py [add/view/
     delete] [task]")
056.         return
057.
058.     command = sys.argv[1].lower()
059.     if command == 'add':
060.         task = ' '.join(sys.argv[2:])
061.         add_task(task)
062.         print("Voce aggiunta.")
063.     elif command == 'view':
064.         view_tasks()
065.     elif command == 'delete':
066.         if len(sys.argv) == 3:
067.             task_number = int(sys.argv[2])
068.             delete_task(task_number)
069.         else:
070.             print("Sintassi: python todo.py delete
     [task number]")
071.     else:
072.         print("Comando invalido. Usa add, view,
     o delete." )
073.
074. if __name__ == "__main__":
075.     main()
```

Custodie per Raspberry Pi 5 – Test di Gruppo

Phil King prova alcune delle migliori custodie disponibili per Raspberry Pi 5

Raspberry Pi 5 è un fantastico computer a scheda singola, ma può essere ancora più figo, all'interno di una custodia intelligente. Ancora più importante, il case può anche aiutarlo a mantenere la scheda fresca sotto un carico di lavoro pesante, utilizzando cuscinetti termici, dissipatore di calore, ventola o una combinazione di tutti e tre. Per testare l'efficacia del raffreddamento, abbiamo eseguito uno stress test utilizzando tutti e quattro i core della CPU con il massimo di default di 2400 GHz. Abbiamo anche controllato il segnale Wi-Fi per vedere se interferivano con esso. Inoltre, abbiamo valutato la capacità di ciascun case – per il montaggio di un HAT ecc. – e l'accesso alle porte del Raspberry Pi 5. Chi è migliore?

Case Ufficiale Raspberry Pi 5

Raspberry Pi | magpi.cc/case
10€ / 10$

Disponibile in rosso/bianco o nero, la custodia ufficiale è composta da tre sezioni di plastica che si agganciano a scatto, rendendo il montaggio semplicissimo. La sezione centrale comprende un inserto trasparente con ventola che si collega alla porta dedicata di Raspberry Pi 5. È incluso anche un piccolo dissipatore di calore per il SoC.

Capacità: Rimuovendo l'inserto trasparente e la ventola dalla sezione centrale, è possibile montare un HAT. In alternativa, è possibile utilizzare un connettore di prolunga per sollevarlo sopra la ventola.

Accessibilità delle porte: Tutte le porte laterali sono accessibili, così come il pulsante di accensione. La sezione centrale presenta un'ampia fessura per l'accesso ai pin GPIO quando il coperchio viene rimosso; potrebbe essere utilizzata anche per il passaggio dei cavi per videocamera o altri accessori.

Raffreddamento: La ventola a velocità variabile viene aiutata dalle prese d'aria nella parte inferiore della custodia, oltre al dissipatore di calore del SoC. Funziona anche a coperchio chiuso, grazie a un foro circolare. Nei nostri test, ha impedito il throttling anche sotto carico intenso, ma altre custodie hanno ottenuto risultati migliori.

Segnale Wi-Fi: Come ci si aspetterebbe, la custodia in plastica ha molto poco effetto sul segnale, che è rimasto forte.

Verdetto

Un design ben congegnato e versatile che copre la maggior parte degli usi.

ICE Tower CPU Cooler

The Pi Hut | magpi.cc/icetower
13€ / 17$

Il suo design a torre dall'aspetto accattivante è arricchito da LED RGB che lo fanno brillare in vari colori. È davvero un case? Bene, ha una base in plastica, collegata tramite bulloni a due distanziali metallici alla sezione ventola/dissipatore di calore nella parte superiore. Il montaggio è un po' complicato, ma non troppo. Il dissipatore di calore è collegato al SoC tramite un pad termico.

Capacità: non potrai montare nessun HAT con questa grande ventola e dissipatore di calore piazzati sul Pi!

Accessibilità delle porte: il design aperto significa accesso a tutte le porte e i pin GPIO non hanno ostacoli.

Raffreddamento: la ventola montata verticalmente soffia aria sulle alette in alluminio del grande dissipatore, la cui base è collegato al SoC del Raspberry Pi 5 tramite un pad termico. Questo si traduce in eccellenti prestazioni di raffreddamento. Per default, la ventola non entra in funzione finché non raggiunge la temperatura di 60°C (cosa che non si è verificata durante i nostri test), ma potrebbe rivelarsi più utile se stai overclockando Raspberry Pi 5.

Segnale Wi-Fi: la presenza di quel dissipatore di calore in metallo ha effetto sul segnale, ma non troppo.

Verdetto

Con il suo design e l'illuminazione insoliti, sembra molto bello... e offre un ottimo raffreddamento.

Argon NEO 5

Argon 40 | **magpi.cc/argonneo5**
18€ / 19$

Un design in tre pezzi – con due sezioni in alluminio e una base in plastica: il NEO 5 trasuda stile e qualità con la sua combinazione di colori rosso/nero e la sensazione di solidità. La sezione centrale è particolarmente impressionante, con una ventola e alette curve per favorire il raffreddamento. Il montaggio è assistito da una guida in un piccolo libretto.

Capacità: Rimuovendo la parte superiore è possibile montare a un HAT con prolunga per il GPIO per sollevarlo sopra la ventola. C'è anche una versione speciale del case per NVMe.

Accessibilità delle porte: la sezione centrale presenta vani per le porte principali, fotocamera/display, PCIe, UART, RTC e pin GPIO (con un'utile etichetta sul lato).

Raffreddamento: l'effetto della ventola è aiutato dalle prese d'aria al centro della sezione e base, più un paio di pad termici per SoC e PMIC. Ciò si traduce in ottime prestazioni di raffreddamento.

Segnale Wi-Fi: il case prevalentemente in metallo comporta una notevole riduzione della potenza e della qualità del segnale.

Verdetto
Un case elegante e di qualità con ottime prestazioni di raffreddamento.

Case KKSB HAT

The Pi Hut | **magpi.cc/kksbhat**
13€ / 16$

Il case più alto del gruppo, è da 56 mm, ed è progettato per ospitare un Raspberry Pi 5 con sopra montato un HAT. La custodia in alluminio anodizzato non ha raffreddamento integrato, ma potresti aggiungere un Active Cooler o un dissipatore. C'è molto spazio e viene fornita una prolunga per il connettore GPIO se necessaria per sollevare leggermente un HAT. Il montaggio è complicato: è necessario rimuovere entrambi i pannelli laterali e far scorrere il Raspberry Pi, avvitarlo in posizione con piccole viti, quindi aggiungere un HAT in seguito.

Capacità: Il grande spazio libero in altezza significa abbondante spazio per montare HAT sul Raspberry Pi. Puoi mettere cose dentro e fuori dalla custodia e fissarle in posizione, è complicato, però. Avremmo preferito un coperchio rimovibile.

Accessibilità delle porte: le porte principali sono accessibili tramite appositi fori. Sono presenti slot per flat cable della fotocamera e PCIe. Dovrai rimuovere uno o entrambi i pannelli laterali per accedere alle altre porte.

Raffreddamento: non è presente un raffreddamento integrato, ma il KKSB può essere utilizzato con un Active Cooler o con la maggior parte degli altri dispositivi di raffreddamento e dissipatori di calore. Le molte fessure nella custodia aiutano la ventilazione.

Segnale Wi-Fi: forse sono tutte quelle fessure di ventilazione, ma... il segnale è difficilmente influenzato.

Verdetto
Non è il design più user-friendly e non ha raffreddamento integrato, ma buona una buona capienza.

ABS Fan Case

The Pi Hut | magpi.cc/absfancase

12€ / 16$

Anche se non sembra il case più solido o pesante, ha due sezioni principali con abbondanti prese d'aria e puoi scegliere le strisce colorate da attaccare sul coperchio! Un il dissipatore di calore Armor Lite V5 con ventola si piazza sopra il Raspberry Pi 5 tramite cinque cuscinetti termici.

Capacità: il dissipatore occupa un discreto spazio nel case. Potresti ancora montare un HAT con una prolunga per il connettore GPIO e senza coperchio.

Accesso alle porte: le porte laterali principali sono accessibili tramite fori sagomati. Togliendo il coperchio è possibile accedere a tutte le altre.

Raffreddamento: cinque pad termici collegano SoC, PMIC, RP1, RAM e modulo wireless con il dissipatore integrato con ventola. Considerando questo, le prestazioni di raffreddamento non sono state buono come ci aspettavamo, ma decenti.

Segnale Wi-Fi: c'è qualche effetto sulla forza e qualità, probabilmente a causa del dissipatore di calore in metallo.

Verdetto

Prestazioni di raffreddamento mediocri in un case leggero.

Passive Cooling Open CNC Case

EDATEC | magpi.cc/opencnccase

8€ / 8$

Raspberry Pi 5 è inserito tra le due sezioni del case, senza parti laterali. Ogni pezzo di alluminio scanalato è dotato di diversi cuscinetti termici per favorire il raffreddamento passivo, inclusi quelli per SoC, PMIC e modulo wireless. Le due sezioni del case sono assicurate con lunghi bulloni.

Capacità: grazie al design aperto e all'accesso al connettore GPIO, puoi montare un HAT appena sopra la sezione superiore.

Accessibilità delle porte: senza parti laterali, l'accesso alle porte è senza ostacoli, con ritagli per i pin GPIO, connettore PoE, Porte PCIe e per fotocamera/display, oltre a connettori UART e batteria RTC.

Raffreddamento: il raffreddamento passivo altamente efficace è fornito dai numerosi pad termici e il case in alluminio scanalato. Mantiene il Raspberry Pi 5 fresco (37,8°C) anche con un carico di lavoro pesante.

Segnale Wi-Fi: Il case in metallo comporta evidente riduzione della potenza e della qualità del segnale.

Verdetto

Ottimo accesso alle porte e prestazioni di raffreddamento passivo impressionanti.

Pibow Coupe 5

Pimoroni | **magpi.cc/pibowcoupe5**
10€ / 10$

Come le precedenti edizioni del classico Pibow, questo case comprende più strati acrilici che devono essere assemblati per circondare Raspberry Pi e quindi fissati con quattro lunghi bulloni. È come fare un puzzle 3D. La parte superiore del Raspberry Pi è lasciata aperta, il che, in alcuni casi d'uso, può essere un vantaggio.

Capacità: il design aperto della custodia ti consente di montare qualsiasi HAT tu voglia.

Accessibilità delle porte: le porte USB ed Ethernet si trovano nella parte superiore della custodia, mentre i ritagli consentono l'accesso al resto. i pin GPIO sono completamente accessibili.

Raffreddamento: non c'è un raffreddamento integrato, ma il Pibow può essere utilizzato con l'Active Cooler o dissipatori standard.

Segnale Wi-Fi: come ci si aspetterebbe da un case aperto, potenza e qualità del segnale sono molto buone.

Verdetto
Un case multicomponente, aperto, senza raffreddamento integrato, ma molto versatile.

Case FLIRC Raspberry Pi 5

The Pi Hut | **magpi.cc/flircrpi5**
11€ / 15$

Come i modelli precedenti, questa custodia FLIRC dall'aspetto smart è progettata per l'uso come media center. La sezione principale in alluminio funge anche da dissipatore di calore tramite un pad termico sul SoC di Raspberry Pi. C'è anche un mini pulsante di accensione.

Capacità: non c'è spazio per un HAT all'interno della custodia.

Accessibilità delle porte: le normali porte laterali sono tutte accessibili tramite ritagli, ma quelle interne – come PCIe e Pin GPIO – non lo sono.

Raffreddamento: Il case in metallo agisce come un grande dissipatore di calore con un pad termico sul SoC, fornendo un raffreddamento passivo efficace.

Segnale Wi-Fi: il case in metallo riduce un poco forza e qualità del segnale.

Verdetto
Una custodia smart e robusta per un media center.

Case Passive Cooling CNC

EDATEC | magpi.cc/cnccase
18€ / 19$

Come per l'Open CNC Case, i due pezzi della custodia sono in alluminio; qui mancano le scanalature, ma mantiene i pad termici. La parte inferiore ha anche fori per le porte sui lati. Il risultato è una custodia dall'aspetto elegante che si adatta perfettamente.

Capacità: il vano per il connettore GPIO consente il montaggio di un HAT sopra la parte superiore del case.

Accessibilità delle porte: i fori consentono completo accesso alle porte principali, insieme a pin GPIO, zoccoli PoE, PCIe e porte fotocamera/display e connettori per UART e batteria RTC.

Raffreddamento: Come per la versione Open del case, il numerosi pad termici (per SoC, PMIC, wireless e parte inferiore) forniscono un raffreddamento passivo altamente efficace.

Segnale Wi-Fi: racchiudere completamente Raspberry Pi 5 in un box metallico ha un effetto negativo sul segnale WIFi.

Verdetto
Un case in metallo con un ottimo raffreddamento, ma il segnale Wi-Fi non è eccezionale.

Argon ONE v3

Argon 40 | magpi.cc/argononev3
29€ / 30$

Il case più grande del gruppo, presenta una scheda figlia che si collega a lato del Raspberry Pi 5. Inoltre riporta le porte video e alimentazione su HDMI e USB-C full-size, il suo chip RP2040 gestisce il raffreddamento e l'alimentazione. C'è anche un ricevitore IR da utilizzare con un telecomando (fornito separatamente). Uno sportello magnetico rimovibile nella parte superiore del case protegge i pin GPIO e la loro etichetta descrittiva. È disponibile anche una versione NVMe del case.

Capienza: Nonostante le dimensioni non è possibile montare un HAT standard all'interno della custodia.

Accessibilità delle porte: USB ed Ethernet sono accessibili, così come alimentazione e due porte HDMI full-size.

Raffreddamento: una ventola angolata nella parte superiore è supportata da due pad termici. È necessario installare uno script per controllare la ventola. Le prestazioni di raffreddamento non sono delle migliori, soprattutto sotto stress.

Segnale Wi-Fi: la potenza e la qualità sono solo leggermente inferiori rispetto al case ufficiale.

Verdetto
Una custodia ricca di funzionalità, ideale per un media center o sistema desktop.

Confronto sul raffreddamento

Abbiamo controllato la temperatura tipica della CPU al minimo e sotto stress (con tutti e quattro i core che funzionano a 2400 MHz).

Case	A vuoto (°C)	Stressato (°C)
Case Ufficiale Raspberry Pi 5	47.2	65.9
ICE Tower CPU Cooler	31.2	48.8
Argon NEO 5	28.5	42.2
Case KKSB HAT*	–	–
Case EDATEC Passive Cooling Open CNC	25.7	37.8
Case ABS Fan	45.5	65.3
Pibow Coupe 5*	–	–
FLIRC	33.4	49.9
Case EDATEC Passive Cooling CNC	27.0	40.0
Argon ONE v3	39.5	67.5

* Nessun raffreddamento integrato.

Segnale Wi-Fi

Abbiamo testato la forza e la qualità media del segnale Wi-Fi di Raspberry Pi 5, per ogni custodia.

Case	Forza (dBm)	Qualità (%)
Case Ufficiale Raspberry Pi 5	-43	96
ICE Tower CPU Cooler	-59	73
Argon NEO 5	-59	73
Case KKSB HAT	-48	90
Case EDATEC Passive Cooling Open CNC	-59	73
Case ABS Fan	-57	76
Pibow Coupe 5	-46	93
FLIRC	-59	71
Case EDATEC Passive Cooling CNC	-65	64
Argon ONE v3	-50	86